DICTIONARY LEARNING AND SPARSE CODING FOR UNSUPERVISED CLUSTERING

By

Pablo Sprechmann

and

Guillermo Sapiro

IMA Preprint Series # 2281

(September 2009)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Dictionary Learning and Sparse Coding for Unsupervised Clustering**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Minnesota,Institute for Mathematics and Its Applications,Minneapolis,MN,55455-0436**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**5**	

DICTIONARY LEARNING AND SPARSE CODING FOR UNSUPERVISED CLUSTERING

Pablo Sprechmann and Guillermo Sapiro

University of Minnesota

ABSTRACT

A clustering framework within the sparse modeling and dictionary learning setting is introduced in this work. Instead of searching for the set of centroid that best fit the data, as in k-means type of approaches that model the data as distributions around discrete points, we optimize for a set of dictionaries, one for each cluster, for which the signals are best reconstructed in a sparse coding manner. Thereby, we are modeling the data as the of union of learned low dimensional subspaces, and data points associated to subspaces spanned by just a few atoms of the same learned dictionary are clustered together. Using learned dictionaries makes this method robust and well suited to handle large datasets. The proposed clustering algorithm uses a novel measurement for the quality of the sparse representation, inspired by the robustness of the ℓ_1 regularization term in sparse coding. We first illustrate this measurement with examples on standard image and speech datasets in the supervised classification setting, showing with a simple approach its discriminative power and obtaining results comparable to the state-of-the-art. We then conclude with experiments for fully unsupervised clustering on extended standard datasets and texture images, obtaining excellent performance.

Index Terms— Clustering, sparse representations, dictionary learning, subspace modeling, texture segmentation.

1. INTRODUCTION

In recent years, sparse representations have received a lot of attention from the signal processing community. This is due in part to the fact that an important variety of signals such as audio and natural images can be well approximated by a linear combination of a few elements (*atoms*) of some (often) redundant basis, usually called *dictionaries*. See [1] and references therein for a review.

Sparse modeling aims at learning these non parametric dictionaries form the data itself. Several algorithms have been developed for this task, e.g., the K-SVD and the method of optimal directions (MOD) (see for example [2] and references therein). Recent publications in a wide spectrum of signals and applications have shown that this approach can be very successful, leading to state-of-the art results, e.g., in image restoration and denoising, texture synthesis, and texture classification.

In the classification setting, this class of algorithms learn dictionaries from the labeled training dataset and use the features of the sparse decomposition of the testing signal for classification (see [2, 3, 4] and references therein). One contribution of our work is to extend these classification strategies to the fully unsupervised setting of data clustering.

In this paper we propose an algorithm for clustering datasets that are well represented in the sparse modeling framework with a set of learned dictionaries. The main idea is, given the number of clusters K, we find the optimal K dictionaries for representing the data, and then associate each signal to the dictionary for which the "best" sparse decomposition is obtained.[1] This is achieved by

$$\min_{\mathbf{D}_i, C_i} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \mathcal{R}(\mathbf{x}_j, \mathbf{D}_i), \tag{1}$$

where $\mathbf{D}_i \in \mathbb{R}^{n \times k_i}$ is the k_i-atoms dictionary associated with the class C_i, $\mathbf{x}_j \in \mathbb{R}^n$ are the data vectors, and $\mathcal{R}$ is a function that measures how good the sparse decomposition for the signal $\mathbf{x}_j$ under the dictionary $\mathbf{D}_i$ is. In the general case, different dictionaries may have different number of atoms, k_i might be cluster dependent. This problem is closely related with the k-q-flat algorithm that aims at finding the closest k q-dimensional flats to a dataset [7]. However, there are major differences between the two. In particular, the framework here proposed, following the sparse representation approach, does not assume a pre-defined, or even constant across classes, (q) dimension, resulting in a richer space for representing and clustering the signals.

We propose a measurement $\mathcal{R}$ for the quality of the sparse representation that naturally takes into account both the reconstruction error and the sparseness (complexity) of the representation on the corresponding learned dictionary. In practice this measurement has shown enormous discrimination power. To further show this we performed experiments in the supervised classification setting using labeled data; we first learned a dictionary for each class, and then classify each testing signal according to this measure. This very simple approach gives results comparable with the state-of-the-art for several benchmark datasets.

The proposed clustering algorithm minimizes (1) using a k-means type of approach that learns a dictionary for each cluster and refines it through the iterations. Experimentally, excellent performance is obtained, both on standard datasets and on texture segmentation tasks.

In the unsupervised clustering case, the initialization is very important for the success of the algorithm. Due to the cost associated with the procedure, repeating random initializations is practically impossible. Thus a "smart" initialization is needed. We propose an approach that combines sparse coding with spectral clustering [8].

Similar ideas to the ones here proposed where previously employed for subspace clustering [9, 10], clustering using the so-called ℓ_1-*graph* by Huang and Yan (see description in [11]), and label propagation [12]. In contrast with our proposed dictionary learning framework, these very inspiring approaches all use the data itself as

Work supported by ONR, NGA, ARO, DARPA, and NSF. We thank I. Ramirez, F. Lecumberry, and J. Mairal for very useful discussions and fundamental software.

[1] Note that it is not that each data point belongs to a union of subspaces as for example in [5, 6]. Comparing with block/group sparsity, here a single dictionary (block) is selected per data point, and the point is sparsely represented (subspace) with atoms only from this dictionary.

dictionary, sparsely representing every data point as a linear combination of the rest of the data. Such representation is computationally expensive (virtually unusable for datasets of thousands of points). In addition, the large redundancy and coherence expected from using the data itself as dictionary is prompt to make the sparse coding very unstable, it is well know that such coding techniques strongly depend on the internal coherence of the dictionary. Furthermore, the performance of these methods decreases when the number of clusters grows. We propose as part of our framework a method to bypass this problem that divides the clustering problem into several binary ones. In a natural way, we use the energy function to decide which partition to choose. Such binary division framework is not so natural for these other related clustering methods.

The remainder of this paper is organized as follows: In Section 2 we briefly summarize the main ideas of *sparse coding* and *dictionary learning*. In Section 3 we define the measure $\mathcal{R}$ and analyze its discriminative power. In Section 4 we present the proposed clustering algorithm and in Section 5 the corresponding experimental results for clustering and texture segmentation. Finally, we conclude the paper in Section 6.

2. SPARSE CODING AND DICTIONARY LEARNING

Sparse coding means to represent a signal as a linear combination of a few atoms of a given (often overcomplete) dictionary. Mathematically, given a signal $\mathbf{x} \in \mathbb{R}^n$ and a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$, the sparse representation problem can be stated as

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t. } \mathbf{x} = \mathbf{D}\alpha, \tag{2}$$

where $||\alpha||_0$ is the "ℓ_0-norm"[2] of the coefficient vector $\alpha \in \mathbb{R}^k$, the number of non-zero elements. This problem is NP-hard, thus is commonly approximated substituting the ℓ_1-norm in Equation (2). In the noisy case the equality constraint must be relaxed as well. An alternative to this is then to solve a Lasso-type problem,

$$\min_{\alpha} ||\mathbf{x} - \mathbf{D}\alpha||_2^2 + \lambda||\alpha||_1, \tag{3}$$

where λ is a parameter that balances the tradeoff between reconstruction error and sparsity. It is a well known fact that in general the ℓ_1 constraint induces sparse solutions for the coefficient vectors α. Furthermore, this is a convex problem that can be solved very efficiently using for example the LARS-Lasso algorithm [13]. This alternative has also been shown to be more stable than the ℓ_0 approach in the sense that in the latter, small variations in the input signal can produce very different active sets (the set of non-zero coefficients in α, or selected atoms from $\mathbf{D}$).

Now, what about the actual dictionary $\mathbf{D}$? State-of-the-art results have shown that it should in general be learned from data. Given a set of signals $\{\mathbf{x}_i\}_{i=1\dots m}$ in $\mathbb{R}^n$, the goal is to find a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$ such that each signal in the set can be represented as a sparse linear combination of its atoms. In this work we use an ℓ_1 variation of MOD following [14]. The algorithm learns the dictionary by solving the following optimization problem:

$$\min_{\mathbf{D},\{\alpha_i\}_{i=1\dots m}} \sum_{i=1}^{m} ||\mathbf{x}_i - \mathbf{D}\alpha_i||_2^2 + \lambda||\alpha_i||_1, \tag{4}$$

restricting the atoms to have unit Euclidean norm. The optimization is carried out using an iterative approach that is composed of two (convex) steps: the sparse coding step on a fixed $\mathbf{D}$ and the dictionary update step with a fixed active set.

Dataset	Proposed	A	B	C	SVM	k-NN
MNIST	1.26	3.41	1.05	-	1.4	5.0
USPS	4.14	3.56	4.38	6.05	4.2	5.2
ISOLET	3.27	4.3	3.4	-	3.3	8.7

Table 1. *Error rate (in percentage) for the classification algorithm discussed in Section 3. We present comparisons with recently published approaches. MNIST: (A) is the best reconstructive method presented in [16], while (B) is the best discriminative one. USPS: (A) is the best reconstructive and (B) is the best discriminative methods reported in [16]. (C) is the best result obtained in [17]. ISOLET: (A) is the supervised k-q-flats and (B) is the k-metrics in [18]. We also compare with a SVM with Gaussian kernel and the Euclidean k-NN.*

3. THE SPARSE REPRESENTATION QUALITY $\mathcal{R}$

A common approach when using dictionaries for classification is to train class specific dictionaries using labeled data and then assign each testing signal to the class for which the best reconstruction is obtained [2, 4]. The measure employed for this task is often the reconstruction error, $\mathcal{R}(\mathbf{x},\mathbf{D}) = ||\mathbf{x}-\mathbf{D}\alpha||_2^2$, where α is the optimal coefficient vector in the sparse coding. While this strategy leads to very good results, it does not take into account the actual sparsity of the reconstruction. Suppose that we have two dictionaries for which almost the same reconstruction error is obtained, but one of them requires double the atoms than the other. In such a situation one would rather select the dictionary that gives the sparsest solution (simplest following Akaike's Information Principle [15]), even if the reconstruction error is slightly bigger.

In practice, this problem can be addressed using a small predefined sparsity level L in an ℓ_0 approach. This strategy is not longer valid when the convex relaxation of Equation (3) is employed. In this situation comparing the reconstruction errors alone has little meaning. We propose then to use the actual cost function in the Lasso problem as a measure of performance, as used in the dictionary learning (4), $\hat{\mathcal{R}}(\mathbf{x},\mathbf{D}) = ||\mathbf{x} - \mathbf{D}\alpha||_2^2 + \lambda||\alpha||_1$, where as before α is the optimal coefficients vector. This alternative naturally takes into account both the reconstruction error and the complexity of the sparse decomposition. The reconstruction error measures the quality of the approximation while the complexity is measured by the ℓ_1 norm of the optimal α.

Let $\mathbf{X}_i$, $i = 1, \dots, K$, be a collection of K (labeled) classes of signals and let $\mathbf{D}_i$ be the corresponding dictionaries trained for each of them independently following for example (4).[3] The class $\hat{j}_0$ for a given new signal $\mathbf{x}$ is found by solving $\hat{j}_0 = \underset{j=1,\dots,K}{\operatorname{argmin}} \hat{\mathcal{R}}(\mathbf{x},\mathbf{D_j})$. This procedure is very simple and has only two parameters: the penalty parameter λ and the size of the dictionaries k. Both can be selected via cross-validation.

As a way to evaluate the discriminatory power of the measure just introduced, which will also be used for the proposed unsupervised clustering approach, we test this simple classification method with standard datasets, the MNIST and USPS digit datasets and the ISOLET data that consists of 617 audio features extracted from 200 speakers saying each letter of the alphabet twice. We used in every case the usual training/testing split. In Table 1 we present the obtained results. We compare our results with several much more sophisticated classification algorithms. The results obtained are comparable and sometimes even better. We also compare with the standard Euclidean k-NN and with SVM with a Gaussian kernel. In all

[2] Although this is normally refered as a norm counting the non-zero elements of a vector, it is actually a pseudo-norm.

[3] See also [2] for cross-training.

our experiments we used a penalty parameter $\lambda = 0.1$. The size of the dictionary depends on the number of training samples as well as the intrinsic complexity of the data. For the MNIST we report results for a dictionary with $k = 800$, $k = 300$ for the USPS digit dataset, and $k = 100$ for the ISOLET. In the last case the training sample is very small, making it impossible to choose larger dictionaries.

One could think of using the whole training datasets as a dictionaries for each class as with the approaches mentioned in the introduction [9, 10, 11]. In that case, in all our experiments the error rates obtained are not better than the ones reported in Table 1. Using the data as dictionaries has the disadvantage that the computational cost of the classification is prohibited,[4] and the method is highly susceptible to label errors due to the high coherence of the "dictionary."

4. DICTIONARY LEARNING FOR CLUSTERING

We now proceed to present the main contribution of this paper, namely, extending the dictionary learning and sparse coding frameworks to unsupervised clustering. Given a set of signals, $\{\mathbf{x}_j\}_{j=1\dots m}$ in $\mathbb{R}^n$, and the number of clusters/classes, K,[5] we want to find the set of K dictionaries $D_i \in \mathbb{R}^{n\times k_i}$, $i = 1, \dots, K$, that best represents the data. We formulate this as an energy minimization problem of the form of Equation (1), and use the measure proposed in Section 3,

$$\min_{\mathbf{D}_i, C_i} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \min_{\alpha_{ij}} ||\mathbf{x}_j - \mathbf{D}_i \alpha_{ij}||_2^2 + \lambda ||\alpha_{ij}||_1, \qquad (5)$$

where as before, the atoms of all the dictionaries are restricted to have unit norm. The optimization is carried out iteratively using a Lloyd's-type algorithm solving one problem at a time: *Assignment step:* The dictionaries are fixed and each signal is assigned to the cluster for which the best representation is obtained: $C_{j_0} := \left\{\mathbf{x} : \hat{\mathcal{R}}(\mathbf{x}, \mathbf{D}_{j_0}) \leq \hat{\mathcal{R}}(\mathbf{x}, \mathbf{D_i})\ \forall i = 1, \dots, K\right\}$. *Update step:* The new dictionaries are computed fixing the assignations found in the previous step. This is the dictionary learning problem (4).

The algorithm stops when the relative change in the energy is less than a given constant. In practice few iterations are needed to reach good results. While the energy is being reduced at every step, there is no guarantee of arriving to a global minimum. In this setting, repeated initializations are computationally very expensive, thus a good initialization is required. This is explained next.

4.1. Initial clusterization

The initialization for the algorithm presented in the previous section can be given as a set of K dictionaries or as an initial partition of the data, this is the C_i sets. We propose two closely related algorithms one corresponding to each of these two alternatives. In both cases the main idea is to construct a similarity matrix and use it as the input for a spectral clustering algorithm.

Let $\mathbf{D}_0 \in \mathbb{R}^{n\times k_0}$ be an initial dictionary, e.g., trained to reconstruct the data for the whole (unlabeled) set $X := [\mathbf{x}_1, \dots, \mathbf{x}_m]$. For each signal $\mathbf{x}_j$ we have the corresponding sparse representation α_j, lets define $\mathbf{A} = [\alpha_1, \dots, \alpha_m] \in \mathbb{R}^{k_0 \times m}$. Two signals belonging to the same cluster are expected to have decompositions that use similar atoms. Thus one can measure the similarity of two signals by comparing the corresponding sparse representations. Inversely, the similarity of two atoms can be determined by comparing how many signals use them simultaneously, and how they contribute, in their sparse decomposition. We compute two matrices representing each one of these cases respectively:

Clustering the signals: Construct a similarity matrix $\mathbf{S}_1 \in \mathbb{R}^{m\times m}$, $\mathbf{S_1} := |\mathbf{A}|^T |\mathbf{A}|$.

Clustering the atoms: Construct a similarity matrix $\mathbf{S}_2 \in \mathbb{R}^{k_0\times k_0}$, $\mathbf{S_2} := |\mathbf{A}|\, |\mathbf{A}|^T$.

In both cases the similarity matrix obtained is positive semidefinite and can be associated with a graph, $G_1 := \{\mathbf{X}, \mathbf{S}_1\}$ and $G_2 := \{\mathbf{D}, \mathbf{S}_2\}$, where the data or the atoms are the sets of vertexes with the corresponding $\mathbf{S}_i$ as edge weights matrixes. This graph is partitioned using standard spectral clustering algorithms to obtain the initialization for the algorithm described in the previous section.

As we mentioned before, G_1 is closely related with the ℓ_1-graph. In that case, the weights of the graph are determined using the sparse decomposition of the signals with the data itself as a dictionary. When the number of signals m is large, the computational cost of constructing the similarity matrix is too expensive. Also the spectral clustering algorithm requires the computation of the largest singular values (and corresponding singular vectors), which is also computationally demanding when m is large (although not so demanding if only a few eigenvectors are needed). In the case of G_2, clustering the atoms bypasses these difficulties: the size of $\mathbf{S}_2$ depends only on the significantly smaller size of the initial dictionary k_0. This parameter does not depend on the amount of data, it just needs to be large enough to model it properly, and is often just in the hundreds. Note that the obtained sub-dictionaries may have different cardinalities (different k_i), reflecting different complexities of the associated clusters.

When the number of clusters, K, is large, the performance of the initial clusterization decreases. We propose a more robust initialization. Starting with the whole set as the only partition, at each iteration we subdivide in two sets each of the current partitions, keeping the division that produces the biggest decrease in the cost energy defined in Equation (5). The procedure stops when the desired number of clusters is reached. This can be applied for any of the two graphs presented in this section, and such partition is consistent with the energy driving the clustering.

Finally, let us make an important observation. Let's consider the ideal situation in which every signal in the K clusters can be exactly reconstructed as a sparse linear combination of the atoms of a dictionary, and that the subspace that they span (using all the atoms) are independent. Assume that the initial dictionary is composed by K (redundant) sub-dictionaries, $\mathbf{D}_0 = [\mathbf{D}_1, \dots, \mathbf{D}_K]$, one corresponding to each cluster in the dataset. Then, given a signal $\mathbf{x}$ belonging to one of them, it is easy to show that the optimal α in the ℓ_1-relaxation of problem (2) with this $\mathbf{D}_0$, will use only atoms from the correct block of the initial dictionary, producing K connected components in both graphs $\mathbf{G}_1$ and $\mathbf{G}_2$. In that situation a spectral clustering technique will successfully separate the clusters. A proof of a similar result is presented in [10].

5. CLUSTERING RESULTS

We now apply the proposed classification algorithm to several clustering problems and texture segmentation. We clustered the digits form 0 to 4 ($K = 5$) using the testing set of MNIST and the training set of USPS. We also clustered the last six letters of ISOLET, $K = 6$, combining the standard training and testing sets.

For the USPS and the MNIST, we used an initial dictionary of $k_0 = 500$ atoms, and $k_0 = 560$ (80×7) for ISOLET. We used G_1 for

[4] With our method, there is the cost of learning the dictionaries, but this is only performed once off-line before the classification.

[5] When K is over-estimated, a micro-detailed partition is observed.

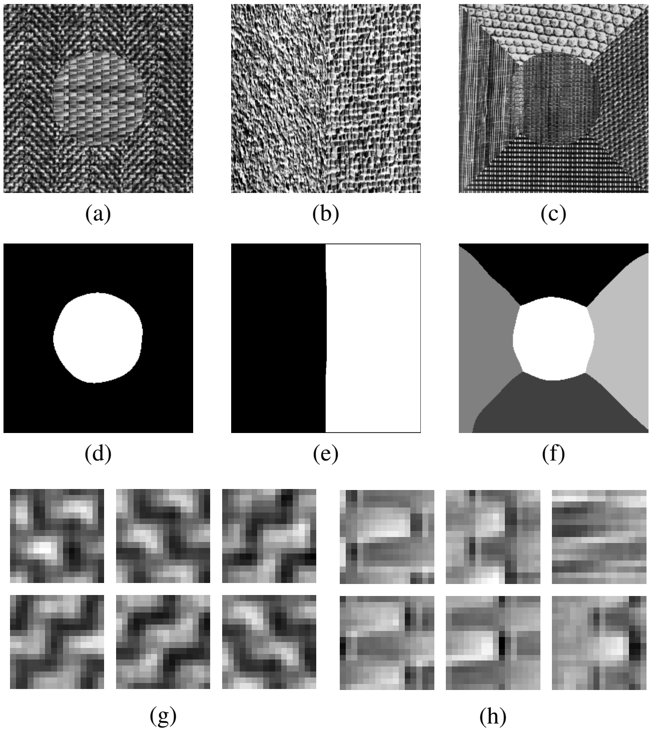

Fig. 1. Results obtained for texture segmentation using the proposed algorithm. The images (a)-(c) are mosaics from the Brodatz database. The obtained results are shown in figures (d)-(f), having 1.74%, 0,25% and 4.25% of misclassified pixels respectively (such misclassifications appear at the regions boundaries, corresponding patches include class mixtures). In images (g) and (h) we show selected atoms of the final sub-dictionaries obtained for image (a). The texture in the circle required $k_1 = 82$ atoms, while the other one received $k_2 = 118$, which goes along with the intuition of larger complexity for this texture. The dictionary learned in the initialization had $K \times 100$ atoms, where again K is the number of textures (clusters) in the image.

initialization, using during the iterations dictionaries of 200 and 100 atoms respectively. In all the cases it was easy to identify the clusters with one of the classes. For the MNIST we had a misclassification rate of 1.44%, for the USPS we obtained 1.6% misclassification (and 7.2% for digits 0-8), and 13% for ISOLET. In the last case most errors where confusions between the letters U-W and T-Z, which have very similar sounds. Note that these results are overall not far from those obtained with supervised learning and classification.

Finally, we use our clustering algorithm for the texture segmentation problem. The approach is related to the one used in [4] for the supervised case. Overlapped 16×16 patches were extracted from the original images and used as input signals to our algorithm. Since the borders on the mosaic images are soft, before each iteration (thus, before recomputing the dictionaries), we applied a Gaussian filter to smooth the segmented regions. In Figure 1 we show some of the results. The number of patches extracted was on the order of several thousands, so the initialization with G_2 was applied. The algorithm gave sub-dictionaires that have a cardinality that intuitively reflects the complexity of the corresponding texture (in other words, k_i was not constant). We got very low rates of miss-clustered pixels, for example in image (b) we got 0.25% which is better than the 0.37% obtained in [2] for the supervised case (which was, as far as we know, the best reported result in the literature for that image).

We observed that best results are obtained for all the experiments when the initial dictionaries in the learning stage are constructed by randomly selecting signals from the training set. If the size of the dictionary compared to the dimension of the data is small, is better to first partition the dataset (using for example Euclidean k-means) in order to obtain a more representative sample.

6. CONCLUDING REMARKS

A framework for unsupervised clustering based on dictionary learning and sparse representations was introduced in this paper. The basic idea is to simultaneously learn a set of dictionaries that optimally represent each one of the clusters. Toward this goal, we introduced a new measurement of representation quality and an initialization procedure that combines sparse coding, dictionary learning and spectral clustering. While here we concentrated on hard clustering, soft-clustering can be obtained as well in this framework.

We are currently pursuing this work in a number of directions, including the incorporation of group incoherence terms [5, 19].

7. REFERENCES

[1] D. L. Donoho A. M. Bruckstein and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Discriminative learned dictionaries for local image analysis,” in *CVPR*, 2008.

[3] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in *NIPS*, vol. 19. 2007.

[4] G. Peyré, “Sparse modeling of textures,” *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, May 2009.

[5] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Trans. IT.*, 2009, to appear.

[6] Z. Zhou A. Ganesh and Y. Ma, “Separation of a subspace-sparse signal: Algorithms and conditions,” in *ICASSP*, april, vol. 14.

[7] P. Tseng, “Nearest q-flat to m points,” *J. Optim. Theory Appl.*, vol. 105, no. 1, pp. 249–252, 2000.

[8] A. Y. Ng, M. Jordan, and Yair Weiss, “On spectral clustering: Analysis and an algorithm,” in *NIPS*, vol. 14. 2002.

[9] S. R. Rao, R.Tron, R. Vidal, and Y. Ma, “Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories,” in *CVPR*, 2008.

[10] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in *CVPR*, 2009.

[11] J. Wright, Y. Ma, J. Mairal, G. Spairo, T. Huang, and S. Yan, “Sparse representations for computer vision and pattern recognition,” in *Proceedings IEEE*, 2009, to appear.

[12] H. Cheng, Z. Liu, and J. Yang, “Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories,” in *ICCV*, 2009.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *ICML*, 2009.

[15] H. Akaike, “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, vol. 19, 1974.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Supervised dictionary learning,” in *NIPS*, vol. 21, pp. 1033–1040. 2009.

[17] K. Huang and S. Aviyente, “Sparse representation for signal classification,” in *NIPS*, 2006.

[18] A.D. Szlam and G. Sapiro, “Discriminative k-metrics,” in *ICML*, 2009.

[19] I. Ramirez, F. Lecumberry, and G. Sapiro, “Universal priors for sparse modeling,” in *IMA Preprint, http://www.ima.umn.edu/preprints/aug2009/2276.pdf*, August 2009.